Docket: 50277-2319 (OID-2002-164-01)   *Patent*

UNITED STATES PATENT APPLICATION

FOR

GENERATING MERGED DOCUMENTS

INVENTORS:

OSAMA ELKADY
TOMOJI ASHITANI

PREPARED BY:

GABRIELA BUTTZ
ORACLE CORPORATION

HICKMAN PALERMO TRUONG & BECKER LLP
1600 WILLOW STREET
SAN JOSE, CALIFORNIA 95125
(408) 414-1080

ASSIGNEE:

ORACLE INTERNATIONAL CORPORATION
500 ORACLE PARKWAY
REDWOOD SHORES, CA 94065

"Express Mail" mailing label number: EV323351051US

Date of Deposit: December 10, 2003

GENERATING MERGED DOCUMENTS

RELATED APPLICATIONS

[0001]   This application claims priority benefit of U.S. Provisional Patent Application, S. N. 60/435,371, filed December 19, 2002, which is hereby incorporated by reference in its entirety.

FIELD OF THE INVENTION

[0002]   The present invention generally relates to electronic documents, and more specifically, to generating electronic documents.

BACKGROUND OF THE INVENTION

[0003]   With the advent of laser printing, many users desire the ability to overlay text data atop background image data. For example, users desire background images, such as company logos, watermarks, headers and footers with an overlay of text based data with character sets, such as data contained in financial and inventory reports, invoices, and checks.

[0004]   A conventional technique for overlaying text onto background templates for reports is to hire consultants to implement a software solution for customizing all of a company's reports. To create these customized reports, the consultants manually convert all of the text reports to include background images, such as a company logo or watermark. Typically, for a suite of reports for a typical financial system, a team of consultants might need to customize as many as 3000 reports. For example, a customer might need to modify thousands files to simply add a company logo to company invoices, checks, and other reports. The modification would need to be to repeated every time a patch or upgrade is

applied to the software. There are some third party solutions to assist customers that have customization requirements. However, third party solutions are often proprietary, may not be supported, may require additional licensing fees, and may add other expenses, such as the expense of hiring consultants that specialize in the third party solution. Some drawbacks to conventional techniques are that hiring consultants and customizing are expensive and time-consuming. Another drawback of conventional techniques is that when a new version of software is released, resources are needed to upgrade customized reports for the newer versions of the software. Yet another drawback to conventional techniques is that many customers choose to not upgrade their software because of the time and effort and expense required for reformatting customized reports.

[0005]   Typically, reports are designed with matching column lengths for data and fixed width columns. A drawback to conventional techniques for reformatting customized reports is that formatting changes for reports must be done manually for each report. Furthermore, including a background image such as a company logo, watermark, header, or footer changes the formatting of the report. The manual formatting and the changes to the formatting add additional expenses and time to the process of generating documents.

[0006]   Another conventional technique for overlaying text onto background templates includes purchasing customized stationery with logos, watermarks, background images, headers, and footers. For printing invoices or checks on high fidelity papers, customers have to use preprinted stationery or use a costly hardware solution that is proprietary. In order to print customized reports, invoices and checks, customers manually feed the customized stationary into a printer. A drawback to customized stationary is that end users need to arrange the position of the report to be printed on the customized stationery by changing characteristics such as margins, pitches, and spaces between lines for every printer.

-2-

Another drawback to customized stationary is that it requires knowledge of printer control commands. Often, the printer control commands vary among printers. In addition, the report has to be reprinted every time a positional attribute is changed so that the report fits in the correct position on the stationery. Furthermore, creating customized stationary requires using specialized printing materials to print invoices, reports, and checks. Another drawback to customized stationary is that it is very expensive.

[0007] Another conventional technique for overlaying text onto background templates is to implement a hardware solution. Hardware solutions typically send escape sequences for layered data and images. A drawback to hardware solution is that hardware solutions typically draw layers one at a time, and the hardware solutions are often report-specific, printer-specific, and hardware-specific. For example, printing different types of customized reports requires different custom hardware solutions. Thus, invoices would require one custom hardware solution, and checks would require another custom hardware solution. Furthermore, each hardware solution may also require a proprietary printer format.

[0008] The present inventors have recognized the above problems and concluded that it is desirable to provide an improved mechanism for overlaying text onto background templates.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009]   The present invention is depicted by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

[0010]   FIG. 1 is a block diagram that depicts a high level overview of a merge and print utility system for the generation of an electronic document, according to an embodiment of the invention.

[0011]   FIG. 2 is a block diagram that depicts a system that includes a merge utility, which may also be used in the merge and print utility system of FIG. 1, according to an embodiment of the invention.

[0012]   FIG. 3 is a flow diagram that depicts a method of operating the merge and print utility system of FIG. 1, according to an embodiment.

[0013]   FIG. 4 is a block diagram that depicts a computer system upon which embodiments of the invention may be implemented.

DETAILED DESCRIPTION OF THE INVENTION

[0014]   A method and apparatus for merging documents is described. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention.

[0015]   The present method and apparatus for merging documents provides several features that can each be used independently of one another or with any combination of the other features. However, any individual feature may not address any of the problems discussed above or may only address one of the problems discussed above. Some of the problems discussed above may not be fully addressed by any of the features described herein. Although headings are provided, information related to a particular heading, but not found in the section having that heading, may also be found elsewhere in the specification. In the respective descriptions of each of FIGs. 1 and 2, all of the components are first listed by name and label number, and then afterwards described in greater detail in later paragraphs.

OPERATIONAL OVERVIEW

[0016]   In an embodiment, a mechanism is provided for overlaying text, graphics, or other items onto background templates in a format that is independent of various printing architectures. In an embodiment the mechanism includes a merge utility for generating electronic documents. In an embodiment, the user may generate output from an arbitrary textual, graphical, or other input coming from any program and can merge the generated output with a background template document.

Docket: 50277-2319 (OID-2002-164-01)

[0017]   For example, a merge utility conversion engine may convert an overlay input data

file into a Standard Printing and Imaging Format (SPIF), such as a Page Description

Language (PDL), a file formatted in the Printer Control Language (PCL) defined by the

Hewlett-Packard Company, or a file formatted in a similar language by any other supplier

such as Epson or Cannon, for example. An example of a PDL is PostSript. Thus, in this

specification, the terms SPIF and PDL are not limited to any particular set of printer

commands, but include similar page description languages.

[0018]   According to an embodiment, the merge and print utility system reads a

background template input data file. The background template input data file contains

information for creating a background for a document, such as a form for a report.

Conversion inputs are sent to an application for providing overlay data, which will be

referred to as an overlay data authoring application. The overlay data is data for adding text,

graphics, or other items to the background associated with the background template input

data. For example, the overlay data may be a company logo, watermark, a color pattern for

the page, a letterhead, a family shield, and/or other item that is later added to the background

template input data. The conversion inputs are processed at the text overlay data authoring

application.

[0019]   At runtime, once the conversion inputs are read, a merge utility of the merge and

print utility system merges the input information to merge the pages of a document or an

overlay and a background into a single document (which may be referred to as a "merged

document"). In an embodiment, the merged document can be used as a master template for

all printed reports. For example, master templates can be re-used for corresponding invoices

and checks based on a particular report. Alternatively, the report may be a form having all

entries filled-in, and is not reusable as a template. According to an embodiment, the merge

Docket: 50277-2319 (OID-2002-164-01)

utility processes the converted overlay data and background template data file inputs and merges the input data into a single SPIF, such as a PDL PostScript file.

[0020]   In an embodiment, a merge utility receives a merge document conversion request for document information that will define a particular output document. The document information may be received in a variety of formats, with each format having a set of commands that define how the overlay data and background template data objects are to be printed for each page of the document. For example, the background template data may be in the form of a text file produced by a particular authoring program, such as Oracle Reports file. The overlay data may be in the form of a file produced by a particular application, such as a Microsoft Word file. Alternatively, the text overlay and background template data document information may be files formatted in a PDL format.

[0021]   The merge and print utility system is not limited to a particular printing device, on a particular command, or based on a particular document type. The merge and print utility generates output that is not limited to any particular printer format file. For explanation purposes only, certain embodiments shall be described in terms of text or other overlay data and background template input data merged in a merge and print utility, which generates output using a standard printer file format, such as the PDL PostScript file format.

A MERGE AND PRINT UTILITY SYSTEM

[0022]   FIG. 1 is a block diagram of a merge and print utility system 100, according to an embodiment of the invention. Merge and print utility system 100 includes a single document 101 produced by merge utility 102, overlay input data 104, background template input data 106, one or more computing devices 108 and 110, one or more output devices 112 and 114, and network connection 122. In alternative embodiments, merge and print utility system100

may not have all of the components listed above or may have other components in addition to or instead of those listed above. The components listed above are discussed in the paragraphs that follow in their numerical order.

[0023]   Single document 101 is single document formed from a merger of two documents that may be of different types. Single document 101 may be in a format, such as SPIF, that can easily be handled by a variety of output devices (a merge format).

[0024]   The merge utility 102 may be referred to as a merge and print utility and may be a computer system with one or more software components or processes operating in a computer system. Merge utility 102 merges two or more documents into single document 101. In an embodiment, the documents being merged are either already in or are placed into the format of single document 101 prior to merger. Merge utility 102 may include a conversion utility to convert documents from an original format into the merge format. Additionally, merge utility 102 may be capable of (1) inspecting a document, (2) determining the application that created the document, and/or (3) sending one or more instructions to the application that created the document to create a version of the original document that is in the merge format.

[0025]   Overlay input data 104 and background template input data 106 represent different types of document data. Overlay input data 104 and background template input data 106 are types of data that a user desires to overlay one on top of the other, and may refer to either or both the versions or merge format versions of the overlay input data 104 and background template input data 106. Overlay input data 104 and background template input data 106, once in the merge format, are converted by merge utility 102 into single document 101. Overlay input data 104 is input data related to an overlay, which may be textual or graphical in nature, for example. Some examples of overlay input data 104 is data related to a

-8-

logo, watermark, or background color pattern. Background template input data 106 may be textual or graphical background data, such as a report or form, which may be generated from another program and in a different format than overlay input data 104. In an embodiment, overlay input data 104 is initially sent to merge utility 102 in its original format, and later sent to merge utility 102 in the merge format. In an embodiment, background template input data 106 is sent to merge utility 102 only in its original format, and later converted by merge utility 102 in the merge format. However, in alternative embodiments, overlay input data 104 and background template input data 106 may be replaced with any two types of data, and may both be sent to merge utility 102 in their, respective, original formats, merged formats, or both. Although only in the examples given in the specification only two documents are merged, in alternative embodiments any number of documents of any number of different types could be merged.

[0026]   Computing devices 108 and 110 may each be a workstation, a personal computer, or other device that can generate a document. Computing devices 108 and 110 may be two different devices, different applications running on the same device, or the same application and device. Computing devices 108 and 110 generate original background and overlay document, which are used to generate overlay input data 104 and background template input data 106, respectively. Computing devices 108 and/or 110 may send overlay input data 104 and background template input data 106 as original documents to merge utility 102, which converts the original documents into converted versions of overlay input data 104 and background template input data 106, respectively. In this specification the terms "converted" and "generated" may be interchanged with one another. Alternatively computing devices 108 and/or 110 receive instructions from merge utility 102 to use original documents created by computing devices 108 and/or 110 to generate converted versions of overlay input data 104

-9-

and/or background template input data 106. For example, computing device 108 may receive instructions to convert an original overlay document into a converted version of overlay input data 104, while computing device 110 may send an original background document to merge utility 102 in the merge format, which in turn generates a converted version of background template input data 106 in the merge format.

[0027]   Output devices 112 and 114 are resources or output devices that can be used to print the pages of a document or store a page of a document in a file, for example. Either of both of output devices 112 and 114 may print single document 101. Output devices 112 and 114 may be different types of devices that handle different output formats, such as two different types of printers. A variety of output devices such as printing devices and computer systems can be used in printing or storing the pages of a document. Thus, the invention is not limited to any particular type of printing or other output device. For example, output devices 112 and 114 may be a combination of color printers, black and white printers, plotters, copiers, and other devices that are used for printing pages of a document.

[0028]   Network connection 122 may be a bus or a computer-readable medium, which may be any one of or any combination of optical, magnetic electrical, mechanical, or acoustic systems for transferring data or signals. Overlay input data 104 and background template input data 106 may be sent to the merge utility 102 and single document 101 may be sent to output device 112 and/or 114 using a variety of methods. For example, overlay input data 104 and background template input data 106 may be sent to the merge utility 102 and single document 101 may be sent to output device 112 and/or 114 via a network connection 122. Alternatively, overlay input data 104 and background template input data 106 may be stored in a database, a mass storage device, or a computer-readable medium, and then loaded onto merge utility 102 separately from the database, the mass storage device, or the computer-

readable medium, for example. In this specification, a database may be a collection of data and software programs to manipulate the data, for example. The database may maintain data through variety of modification operations, such as insert, update, and delete operations.

MERGE UTILITY

[0029] FIG. 2 depicts system 200 having merge utility 102, which may include server 202, produce conversion instructions 203, and include merge utility conversion engine 204, which in turn may include merge utility conversion engine interface 206. The merge utility 102 interacts with original overlay input data 104a, converted overlay input data 104b, original background template input data 106a, and converted background template input data 106b, and document authoring applications, such as background template document authoring application 208 and overlay document authoring application 212. According to an embodiment, background template document authoring application 208 and overlay document authoring application 212 include a subset of the functionality of the application used to create the original documents, and also include document authoring application interfaces 210 and 214, respectively. System 200 also includes document merge request 230, document conversion response 240, output device 250, and conversion utility 280. In alternative embodiments, system 200 may not have all of the components listed above or may have other components in addition to or instead of those listed above. The components of system 200 are described in the order of the flow of information starting from inputs document merge request 230, background template document authoring application 208, and overlay document authoring application 212 and ending with output device 250.

[0030] Document merge request 230 may be a request to merge a document with a background before printing or storing. According to an embodiment, document merge

request 230 may be a request to merge or a request to both merge and print documents (a merge and print conversion request) and may comprise document conversion parameters, or other document information such as document location information, or the name of the document authoring application that created the original document. In an embodiment, the merge request 230 includes information about an editor and conversion mechanism, the environment for the conversion of the original document to a PDL format or other SPIF.

[0031]   Regarding background template document authoring application 208, original background template input data 106a and converted background template input data 106b are background template input data 106 in the format in which it was originally created and in the merge format to which it is later converted, respectively. Background template authoring application 208 may run on computing device 110, and in an embodiment, produces original background template data document (e.g., background template input data 106A) in an original format that later becomes the background of the merged document. Original background template input data 106a may be generated, at background template document authoring application 208, using a widely available WYSIWIG document authoring application, such as Microsoft PowerPoint, Microsoft Word, or Microsoft Excel. Background template document authoring application interface 210 passes the background template input data 106a from background template authoring application 208 to conversion utility 280.

[0032]   Conversion utility 280 converts the background template (original background template input data 106a) from an original format to converted background template input data 106b in the merge format supported by output device 250. The merge format may be in the same format as the merged document (single document 101), except that the background and overlay have not yet been merged. Alternatively, the merge format may be a different

format than the merged document. Conversion utility 280 may be incorporated within merge utility 102 or may be a separate module. Conversion utility 280 may be a widely available driver, such as Xpdf pdftops for Microsoft software and an Adobe generic printer driver utility to generate a SPIF file, such as a PDL PostScript file.

[0033] Regarding overlay document authoring application 212, original overlay input data 104a and converted overlay input data 104b are background template input data 104 in the format in which it was originally created and in the merge format to which it is later converted, respectively. Overlay document authoring application 212 may run on computing device 108, and produces overlay data documents (e.g., overlay input data 104) that govern how the overlay document is overlayed on the background. In an embodiment, overlay document authoring application 212 produces overlay input data 104 in both an original and merge format, as original overlay input data 104a and converted overlay input data 104b. In an embodiment, overlay document authoring application interface 214 facilitates providing overlay input data 104b to merge utility 102 in a format that merge utility 102 can handle. Specifically, in an embodiment, overlay document authoring application interface 214 receives instructions, based on merge request 230, that are executed by overlay document authoring application 212 for converting original overlay input data 104a from an original to the merge format. Conversion instructions 203 are one or more instructions received by overlay document authoring application 212 that cause overlay document authoring application 212 to generate converted overlay input data 104b.

[0034] In an embodiment, merge utility 102 may be a merge and print utility for producing a merged SPIF document from a converted background template input data document 106b (from background template document authoring application 208) and overlay input data 104b (from overlay document authoring application 212) based on document

-13-

merge request 230. According to an embodiment, merge utility 102 receives document merge request 230, retrieves a background template data document (e.g., background template input data 106b) and a text overlay data document (e.g., overlay input data 104b) and produces a merged SPIF document (e.g., single document 101). Merge utility 102 may store the merged SPIF document as a template for later use. Merge utility 102 replies to the document merge request 230 by sending single document 101 as a document conversion response. The SPIF document, single document 101, produced by merge utility 102 preserves the visual presentation properties or virtually all of the visual presentation properties of the original overlay document (produced by overlay document authoring application 212) and background template data document (produced by background template document authoring application 208), as characterized by the document authoring application used to create the original document (overlay document authoring application 212).

[0035]   Within merge utility 102, server 202 processes document conversion requests 230 for merge utility conversion engine 204 and processes the output of merge utility conversion engine 204 as conversion response 240. Server 202 is configured to receive and send document conversion requests and responses to the merge utility conversion engine 204. In an embodiment server 202 may be a separate unit than merge utility 102 rather than being incorporated within merge utility 102. At merge utility 102, server 202 receives document merge request 230. Server 202 passes the document merge request 230 to the merge utility conversion engine 204.

[0036]   Within merge utility 102, merge utility conversion engine 204 compiles document conversion instructions 203 that dictate how the document conversion is to be performed based on document merge request 230 from server 202. In other words, merge utility conversion engine 204 generates conversion instructions 203 based on document

-14-

merge request 230. According to an embodiment of the invention, the conversion

instructions 203 may include information concerning the document authoring application that

created the original document. Merge utility conversion engine 204 performs the actual

merger of overlay input data 104 and background template input data 106 after they have

been converted to a format appropriate for performing the merger, such as a PDL PostScript,

another SPIF or another print-ready or merge format. The merged document may comprise a

single file or document (e.g., single document 101) that retains the visual properties of the

background template data with an overlay of the text overlay data.

[0037] In an embodiment, merge utility conversion engine 204 may include conversion

utility 280 to convert documents from an original format into the merge format.

Additionally, merge utility conversion engine 204 may be capable of (1) inspecting an

original overlay document in its original format, such as original overlay input data 104a, (2)

determining the application that created the document (e.g., overlay document authoring

application 212), (3) sending one or more instructions, as conversion instructions 203, to

overlay document authoring application 212 to create converted overlay input data 104b that

is in the merge format, and/or (4) converting the merge format versions of overlay input data

104 and background template input data 106 into singe document 101. In an embodiment,

overlay document authoring application 212 and overlay input data 104 may be an MS Word

application and an MS Word document, and merge utility conversion engine 204 may

generate conversion instructions 203 that causes a MS Word application to print an MS Word

document into a postscript file.

[0038] Also within merge utility 102, merge utility conversion engine interface 206 acts

as an interface between merge utility conversion engine 204 and text overlay authoring

application 212. Merge utility conversion engine interface 206 also acts as an interface

-15-

between merge utility conversion engine 204 and conversion utility 280. Merge utility conversion engine interface 206 calls overlay document authoring application interface 214, and passes it the conversion instructions 203 compiled by merge utility conversion engine 204, for processing by overlay authoring application 212. Documents from text overlay authoring application 212 and conversion utility 280 may be received by merge utility conversion engine 204 via merge utility conversion engine interface 206. Instructions related to sending and/or converting the format of background template input data 106 may be sent from merge utility conversion engine 204, via merge utility conversion engine interface 206, to background template document authoring application interface 210.

[0039]   Response 240 is the merged document produced by merge utility 102, which may be single document 101. Response 240 is sent by server 202 to output device 250 for printing, for example. Output device 250 may be the same as output devices 112 or 114.

METHOD OF PRODUCING A MERGED DOCUMENT

[0040]   FIG. 3 is a flowchart illustrating a method 300 of producing a merge document from electronically stored original documents, in accordance with an embodiment of the invention.

[0041]   At step 302, server 202 receives document merge request 230. In FIG. 2, merge utility is abbreviated as MU. At step 304, document merge request 230 is passed to merge utility conversion engine 204 for processing. From step 304, method 300 proceeds to steps 309 and 329, which may be performed in any order with respect to each other.

[0042]   At step 309, background template data (e.g., converted background template input data 106b) is created via a number of sub-steps. At sub-step 310 a background template data document is generated (e.g., original background template input data 106a) by

background template document authoring application 208. Sub-step 310 may include generating original background template input data 106a using a widely available WYSIWIG document authoring application, such as Microsoft PowerPoint, Microsoft Word, or Microsoft Excel.

[0043]   At sub-step 312 the original background template data input data 106a is converted to a SPIF file by conversion utility 280 as converted background template input data 106b (which may include executing a widely available driver, such as Xpdf pdftops for Microsoft software and an Adobe generic printer driver utility to generate a SPIF file, such as a PDL PostScript file). According to an embodiment, at sub-step 312 an Xpdf utility may be run on the client side where the background template data document authoring application 208 resides or on the server side where a server side command is invoked to convert the original background template data document to a SPIF file. According to an embodiment, control tags may be added to the merge converted SPIF file to define the merge format. Control tags provide document format and content definitions such as header definition, function definitions, fonts, page contents background file data for the resultant merged file, and document structuring conventions.

[0044]   At sub-step 316, the merge utility conversion engine interface 206 returns the SPIF conversion results (converted background template input data 106b) to the merge utility conversion engine 204 for processing and merging with text overlay data. In an embodiment, sub-step 316 is the last step in step 309.   In alternative embodiments, step 309 may not have all of the sub-steps listed above or may have other sub-steps in addition to or instead the sub-steps listed above.

[0045]   At step 329, overlay data (e.g., overlay input data 104) is created via a number of sub-steps. At sub-step 330, the merge utility 102 retrieves the original overlay document

(e.g., original overlay input data 104a) from the original document authoring application (e.g., overlay document authoring application 212), which is in an original format created by overlay document authoring application 212. Merge utility conversion engine 204 extracts information, from merge request 230 and/or from original overlay input data 104a, that may be used later to convert original overlay input data 104a into the merge format. In an embodiment of the invention, the merge request 230 includes information about an editor and conversion mechanism, the environment for the conversion of the original document to a PDL format document. Original overlay input data 104a may be created using any editor.

[0046]   At sub-step 332, the merge utility conversion engine 204 determines the electronic storage format and/or a presentation format related to original overlay data input 104a. According to an embodiment, the original overlay input data 104a may be stored in a database, a mass storage device, or a computer-readable medium. According to an embodiment, the storage format for original overlay input data 104a may be in a graphic or text format, for example, such as a report generated by a report generator. For example, the report generator may be Hyperion Enterprise, SAP, or Oracle Reports, or some other component of the Oracle family of products. According to an embodiment, the storage format of original overlay input data 104a may be of another type, such as Adobe Photoshop or Adobe Acrobat, or some other component of the Adobe family of products. In an embodiment of the invention, the presentation format and/or storage format of the original document may be indicated within merge request 230 or an element of the merge utility conversion engine 204, for example.

[0047]   At sub-step 334, based on the information extracted from conversion request 204 and/or the document type of original overlay input data 104a, the merge utility conversion engine 204 generates conversion instruction, and passes the conversion instructions to merge

-18-

utility conversion engine interface 206. The conversion instructions may comprise a reference containing an image file name, attributes of the original overlay input data 104a, such as the paper size of the document, and the margin size of the document, and conversion mechanism information. Conversion mechanism information may include, for example, information for conversion of the original document to a merge SPIF, such as PDL PostScript. According to an embodiment, the original documents may be converted to PostScript format either by pdftops command by Xpdf or with an Adobe Generic PostScript printer driver.

[0048]   At sub-step 336, overlay document authoring application 112 generates conversion results, which may be converted overlay input data 104b, from the original overlay documents (original overlay input data 104a). In other words, overlay document authoring application 112 converts overlay input data 104 from an original format to a merge format. The generated conversion results preserve the visual presentation properties of the original overlay input data 104a. At sub-step 338, document authoring application interface 214 returns the conversion results, as converted overlay input data 104b, to the merge utility conversion engine interface 206. In an embodiment, sub-step 338 is the last step in step 329. In alternative embodiments, step 329 may not have all of the sub-steps listed above or may have other sub-steps in addition to or instead the sub-steps listed above.

[0049]   At step 340 the merge utility conversion engine interface 206 returns the conversion results to the merge utility conversion engine 204. At step 346, the merge utility conversion engine 204 merges the conversion results, by for example merging converted overlay input data 104b and converted background template input data 106b into a merge document, such as single document 101. According to an embodiment of the invention, the merge utility conversion engine 204 may produce a single merged document in a format

supported by output devices 112, 114, and 250 (e.g., single file 101), and including the visual presentation properties of the original documents, as determined by the document authoring application that created the original background template document (background template document authoring application 208), and as determined by the overlay document authoring application 212 that created the original overlay document (overlay input data 104).

[0050]   At step 348, the merge utility conversion engine 204 passes information concerning single document 101, as a SPIF output file, to server 202. At step 350, the merge utility conversion engine 204 passes single document 101 to an output device (e.g., output device 112, 114, or 250).

[0051]   As described above, in an embodiment, a computer implemented method 300 combines multiple SPIF files into a single print document (single document 101) for printing and may facilitate customization and maintenance of customized reports. In an embodiment, merge utility 102 converts a overlay input data 104 into a SPIF document internally as well as reads another converted background template input data 106b, as a SPIF file, and generates single document 101 for printing. In an embodiment, merge utility 102 generates output for a variety of textual or other input coming from any program and can merge the generated output with a second background template document. In an embodiment, the passing of single document 101 to output device 112, 114, or 250 and the merging of the converted background template input data 106b with converted overlay input data 104b are executed at printing time.

HARDWARE OVERVIEW

[0052]   The approach for generation of an electronic document described herein may be implemented in a variety of ways and the invention is not limited to any particular implementation. The approach may be integrated into a computing system or a printing device, or may be implemented as a stand-alone mechanism. Furthermore, the approach` may be implemented in computer software, hardware, or a combination thereof.

[0053]   FIG. 4 is a block diagram that depicts a computer system 400 upon which an embodiment of the invention may be implemented. Computer system 400 is an example of a computer system that may be used for computing device 108, computing device 110, and/or merge utility 102. Alternatively, merge utility may be a plurality of computer systems similar to computer system 400 connected together in a network. For example, server 202 may be one computer system 400 and merge utility conversion engine may be another computing system 400. Alternatively, computing device 108, computing device 110, merge utility 102, background template document authoring application 208, and/or overlay document authoring application 212 may be separate applications running on the same computer system 400.

[0054]   Computer system 400 includes a bus 402 or other communication mechanism for communicating information, and a processor 404 coupled with bus 402 for processing information. Computer system 400 also includes a main memory 406, such as a random access memory (RAM) or other dynamic storage device, coupled to bus 402 for storing information and instructions to be executed by processor 404. Main memory 406 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 404. Computer system 400 further includes a read only memory (ROM) 408 or other static storage device coupled to bus 402 for storing static

-21-

information and instructions for processor 404. A storage device 410, such as a magnetic disk or optical disk, is provided and coupled to bus 402 for storing information and instructions.

[0055]   Computer system 400 may be coupled via bus 402 to a display 412, such as a cathode ray tube (CRT), for displaying information to a computer user. An input device 414, including alphanumeric and other keys, is coupled to bus 402 for communicating information and command selections to processor 404. Another type of user input device is cursor control 416, such as a mouse, a trackball, or cursor direction keys for communicating direction information and command selections to processor 404 and for controlling cursor movement on display 412. This input device typically has two degrees of freedom in two axes, a first axis (e.g., x) and a second axis (e.g., y), that allows the device to specify positions in a plane.

[0056]   The invention is related to the use of computer system 400 for implementing the techniques described herein. According to one embodiment of the invention, the method of merging documents (e.g., method 300), described above, are performed by computer system 400 in response to processor 404 executing one or more sequences of one or more instructions contained in main memory 406. Such instructions may be read into main memory 406 from another computer-readable medium, such as storage device 410. Execution of the sequences of instructions contained in main memory 406 causes processor 404 to perform the process steps described herein. In alternative embodiments, hard-wired circuitry may be used in place of or in combination with software instructions to implement the invention. Thus, embodiments of the invention are not limited to any specific combination of hardware circuitry and software.

Docket: 50277-2319 (OID-2002-164-01)

[0057]   The term "computer-readable medium" as used herein refers to any medium that participates in providing instructions to processor 404 for execution, and is an example of a machine-readable medium. The methods of merging documents described herein, such as method 300, may be carried (e.g., stored on) such computer-readable media. Such a medium may take many forms, including but not limited to, non-volatile media, volatile media, and transmission media. Non-volatile media includes, for example, optical or magnetic disks, such as storage device 410. Volatile media includes dynamic memory, such as main memory 406. Transmission media includes coaxial cables, copper wire and fiber optics, including the wires that comprise bus 402. Transmission media can also take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0058]   Common forms of computer-readable media include, for example, a floppy disk, a flexible disk, hard disk, magnetic tape, or any other magnetic medium, a CD-ROM, any other optical medium, punch cards, paper tape, any other physical medium with patterns of holes, a RAM, a PROM, and EPROM, a FLASH-EPROM, any other memory chip or cartridge, a carrier wave as described hereinafter, or any other medium from which a computer can read.

[0059]   Various forms of computer-readable media may be involved in carrying one or more sequences of one or more instructions to processor 404 for execution. For example, the instructions may initially be carried on a magnetic disk of a remote computer. The remote computer can load the instructions into its dynamic memory and send the instructions over a telephone line using a modem. A modem local to computer system 400 can receive the data on the telephone line and use an infra-red transmitter to convert the data to an infra-red signal. An infra-red detector can receive the data carried in the infra-red signal and

appropriate circuitry can place the data on bus 402. Bus 402 carries the data to main memory 406, from which processor 404 retrieves and executes the instructions. The instructions received by main memory 406 may optionally be stored on storage device 410 either before or after execution by processor 404.

[0060]   Computer system 400 also includes a communication interface 418 coupled to bus 402. Communication interface 418 provides a two-way data communication coupling to a network link 420 that is connected to a local network 422. Network link 420 is an example of network connection 122. For example, communication interface 418 may be an integrated services digital network (ISDN) card or a modem to provide a data communication connection to a corresponding type of telephone line. As another example, communication interface 418 may be a local area network (LAN) card to provide a data communication connection to a compatible LAN. Wireless links may also be implemented. In any such implementation, communication interface 418 sends and receives electrical, electromagnetic or optical signals that carry digital data streams representing various types of information.

[0061]   Network link 420 typically provides data communication through one or more networks to other data devices. For example, network link 420 may provide a connection through local network 422 to a host computer 424 or to data equipment operated by an Internet Service Provider (ISP) 426. ISP 426 in turn provides data communication services through the world wide packet data communication network now commonly referred to as the "Internet" 428. Local network 422 and Internet 428 both use electrical, electromagnetic or optical signals that carry digital data streams. The signals through the various networks and the signals on network link 420 and through communication interface 418, which carry the digital data to and from computer system 400, are exemplary forms of carrier waves transporting the information.

-24-

Docket: 50277-2319 (OID-2002-164-01)

[0062]   Computer system 400 can send messages and receive data, including program code, through the network(s), network link 420 and communication interface 318. In the Internet example, a server 430 might transmit a requested code for an application program through Internet 428, ISP 426, local network 422 and communication interface 418.

[0063]   The received code may be executed by processor 404 as it is received, and/or stored in storage device 410, or other non-volatile storage for later execution. In this manner, computer system 400 may obtain application code in the form of a carrier wave.

EXTENSIONS AND ALTERNATIVES

[0064]   In the foregoing specification, the invention has been described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention. Thus, the specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense. The invention includes other contexts and applications in which the mechanisms and processes described herein are available to other mechanisms, methods, programs, and processes.

[0065]   In addition, in this disclosure, certain process steps are set forth in a particular order, and alphabetic and alphanumeric labels are used to identify certain steps. Unless specifically stated in the disclosure, embodiments of the invention are not limited to any particular order of carrying out such steps. In particular, the labels are used merely for convenient identification of steps, and are not intended to imply, specify or require a particular order of carrying out such steps. Furthermore, other embodiments may use more or fewer steps than those discussed herein.